REMARKS

The present Amendment cancels claims 1-8 and adds new claims 9-14. Therefore, the present application has pending claims 9-14.

35 U.S.C. §112 Rejections

Claims 6 and 7 stand rejected under 35 U.S.C. §112, second paragraph as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter of the invention. This rejection is traversed for the following reasons. As previously indicated, claims 6 and 7 were canceled. Therefore, this rejection regarding claims 6 and 7 is rendered moot.

Furthermore, Applicants submit that new claim 13 corresponds to canceled claim 6. Applicants further submit that new claim 13, as clearly recited, is in compliance with the provisions of 35 U.S.C. §112.

35 U.S.C. §103 Rejections

Claims 1-6 and 8 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U. S. Patent No. 4,780,821 to Crossley in view of U. S. Patent No. 5,414,851 to Brice, Jr. et al. ("Brice"). As previously indicated, claims 1-6 and 8 were canceled. Therefore, this rejection regarding claims 1-6 and 8 is rendered moot.

Claim 7 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Crossley, in view of Brice, and further in view of U.S. Patent Application Publication No. 2002/0083226 to Awasthi et al. ("Awasthi"). As previously indicated, claim 7 was canceled. Therefore, this rejection regarding claim 7 is rendered moot.

New Claims 9-14

Claims 9-14 were added to more clearly describe features of the present invention. More specifically, claims 9-14 were added to more clearly describe that the present invention is directed to a computer system having a virtualized input/output (I/O) device as recited, for example, in independent claim 9.

Applicants submit that the features of the present invention, as now more clearly recited in claims 9-14, are not taught or suggested by Crossley, Brice or Awasthi, whether taken individually or in combination with each other in the manner suggested by the Examiner. Therefore, Applicants respectfully request the Examiner to reconsider and withdraw this rejection.

The present invention, as recited in claim 9, provides a computer system having a virtualized input/output (I/O) device. The computer system includes a client computer and a server computer. The client computer includes a first processor, a first hypervisor and a first operating system (OS) operating on the first hypervisor. The server computer includes a second processor, a second hypervisor, a second OS operating on the second hypervisor, and a physical I/O device. According to the present invention, the client computer and the server computer are connected via a network. Also according to the present invention, the first hypervisor includes memory protection interrupt processing means called by a memory protection interrupt that is generated when a read or a write occurs to a particular memory address in the first processor. The first hypervisor also includes logical I/O device access detecting means that is called by the memory protection interrupt processing means to detect a logical I/O device access, which is directed to the physical I/O device of the server computer. Furthermore, the first hypervisor includes a virtual I/O client processing module for transmitting to the server computer, via the network,

a command to access the logical I/O device when the logical I/O device access is detected. In addition, according to the present invention, the second hypervisor includes a virtual I/O server processing module for receiving the command via the network and for issuing a command to the physical I/O device. The prior art does not teach or suggest all of these features.

The above described features of the present invention, as now more clearly recited in the claims, are not taught or suggested by any of the references of record. Specifically, the features are not taught or suggested by either of Crossley, Brice or Awasthi, whether taken individually or in combination with each other.

Crossley teaches a method for multiple program management within a network having a server computer and a plurality of remote computers. However, there is no teaching or suggestion in Crossley of the computer system having a virtualized I/O device as recited in claim 9 of the present invention.

Crossley discloses a procedure which allows users of a computer system, which includes a plurality of computers connected in a local area network, to share both file resources and application programs on the local area network without modification to existing programs which were designed to run in a non-network environment. The local area network includes a server computer and at least one remote computer. Starting the network includes an initial program load of the operating system for each of the computers, loading the local area network control program and then loading a hypervisor or "node enabler" program. At each of the remote computers, a request to load a program or access a data file is converted by the "node enabler" to a file sharing and record locking protocol message which is transmitted to the server computer. The server computer stores a program matrix with entries indicating which programs can be run on the network without conflicts

with other systems including the server computer. The server computer also maintains a list of currently running programs and accessed data files. By comparing the remote computer request with the program matrix and the list of currently running programs and accessed data files, a decision to grant a remote computer's request is made. In addition, by recording a unique identification number for each remote computer signed on to the network at the server computer, control of access to licensed programs is maintained.

One feature of the present invention, as recited in claim 9, includes where the first hypervisor includes memory protection interrupt processing means called by a memory protection interrupt that is generated when a read or a write occurs to a particular memory address in the first processor, logical I/O device access detecting means that is called by the memory protection interrupt processing means to detect a logical I/O device access, which is directed to the physical I/O device of the server computer, and a virtual I/O client processing module for transmitting to the server computer, via the network, a command to access the logical I/O device when the logical I/O device access is detected. Crossley does not disclose this feature.

Another feature of the present invention, as recited in claim 9, includes where the second hypervisor includes a virtual I/O server processing module for receiving the command via the network and for issuing a command to the physical I/O device. Crossley does not disclose this feature.

In the present invention, hypervisors are provided to access the I/O device of a remote computer through a network. Processing is intercepted when called by the memory protection interrupt. More specifically, when writing in a specific memory address or reading is caused on the client computer, a memory protection interrupt is generated (see e.g., page 10, lines 18-22). Memory protection interrupt

processing of the first hypervisor is called, and logical I/O device access detection is performed by the memory protection interrupt processing (see e.g., page 10, lines 2-7; page 12, lines 15-22; and Fig. 6, step 303). After detection of the logical I/O device, a logical I/O device command is forwarded to the server computer by virtual I/O client processing. Finally, I/O is executed by a second hypervisor of the server computer.

As described in column 2, lines 28-33, the Crossley system allows a large base of existing application programs written for a non-network environment to be used in a multi-user/multi-tasking environment without having to rewrite the source code of the application programs. To provide this system, Crossley introduces a hypervisor or "node enable" program to the system (see e.g., the Abstract). The hypervisor of Crossley intercepts all program load/execute requests and file processing requests (see e.g., column 4, lines 25-27). This is quite different from the present invention, where the I/O device access selectively functions by the storage protection interrupt function of the processor hypervisor, and is made virtual.

Therefore, Crossley fails to teach or suggest "wherein said first hypervisor comprises: memory protection interrupt processing means called by a memory protection interrupt that is generated when a read or a write occurs to a particular memory address in said first processor, logical I/O device access detecting means that is called by said memory protection interrupt processing means to detect a logical I/O device access, which is directed to said physical I/O device of said server computer, and a virtual I/O client processing module for transmitting to said server computer, via said network, a command to access said logical I/O device when said logical I/O device access is detected" as recited in claim 9.

Furthermore, Crossley fails to teach or suggest "wherein said second hypervisor comprises a virtual I/O server processing module for receiving said command via said network and for issuing a command to said physical I/O device" as recited in claim 9.

The above noted deficiencies of Crossley are not supplied by any of the other references of record, namely Brice, whether taken individually or in combination with each other. Therefore, combining the teachings of Crossley and Brice still fails to teach or suggest the features of the present invention as now more clearly recited in the claims.

Brice teaches a method and means for sharing I/O resources by a plurality of operating systems. However, there is no teaching or suggestion in Brice of the computer system having a virtualized I/O device as recited in claim 9 of the present invention.

Brice discloses a method for increasing the connectivity of I/O resources to a multiplicity of operating systems (OSs) running in different resource partitions of a computer electronic complex (CEC) to obtain sharing of the I/O resources among the OSs of the CEC, including channels, subchannels (devices), and control units (CUs). Brice provides image identifiers (IIDs) for assigning resources to the different OSs. Each shared I/O resource has a sharing set of control blocks (CBs) in which a respective CB is assigned to (and located by) a respective IID of one of the OSs. Each of the CBs in a sharing set provides a different image of the same I/O resource. The different CB images are independently set to different states by I/O operations for the different OSs, so that the OSs can independently share the same I/O resource.

One feature of the present invention, as recited in claim 9, includes where the first hypervisor includes memory protection interrupt processing means called by a memory protection interrupt that is generated when a read or a write occurs to a particular memory address in the first processor, logical I/O device access detecting means that is called by the memory protection interrupt processing means to detect a logical I/O device access, which is directed to the physical I/O device of the server computer, and a virtual I/O client processing module for transmitting to the server computer, via the network, a command to access the logical I/O device when the logical I/O device access is detected. Brice does not disclose this feature.

Another feature of the present invention, as recited in claim 9, includes where the second hypervisor includes a virtual I/O server processing module for receiving the command via the network and for issuing a command to the physical I/O device. Brice does not disclose this feature.

Although Brice discloses a hypervisor, Brice does not disclose close the use of hypervisors, as claimed. As described in column 2 (last line) to column 3, line 3, Brice merely discloses a function of coordination for dividing a processor resource and/or a memory resource into each operating system that operates individually. Furthermore, the Brice system directly shares all the physical I/O resources without intervention from the hypervisor (see column 6, lines 34-36). This is quite different form the present invention, where a hypervisor is used to make a virtualized I/O device (i.e., access of remote I/O).

Therefore, Brice fails to teach or suggest "<u>wherein said first hypervisor comprises: memory protection interrupt processing means called by a memory protection interrupt that is generated when a read or a write occurs to a particular memory address in said first processor, logical I/O device access detecting means</u>

that is called by said memory protection interrupt processing means to detect a logical I/O device access, which is directed to said physical I/O device of said server computer, and a virtual I/O client processing module for transmitting to said server computer, via said network, a command to access said logical I/O device when said logical I/O device access is detected" as recited in claim 9.

Furthermore, Brice fails to teach or suggest "wherein said second hypervisor comprises a virtual I/O server processing module for receiving said command via said network and for issuing a command to said physical I/O device" as recited in claim 9.

The above noted deficiencies of Crossley and Brice are not supplied by any of the other references of record, namely Awasthi, whether taken individually or in combination with each other. Therefore, combining the teachings of Crossley, Brice and Awasthi still fails to teach or suggest the features of the present invention as now more clearly recited in the claims.

Awasthi teaches configuring computer components. However, there is no teaching or suggestion in Awasthi of the computer system having a virtualized I/O device as recited in claim 9 of the present invention.

Awasthi discloses techniques for configuring network interface cards. The disclosed techniques include storing device information related to multiple network interface card, and configuring the network interface cards based on the stored device information using a device driver. Furthermore, Awasthi discloses techniques for installing a peripheral device. These techniques include initiating a search of stored device information by a device driver, receiving in the device driver information about the peripheral device in response to the search, and configuring the device using the received information.

One feature of the present invention, as recited in claim 9, includes where the first hypervisor includes memory protection interrupt processing means called by a memory protection interrupt that is generated when a read or a write occurs to a particular memory address in the first processor, logical I/O device access detecting means that is called by the memory protection interrupt processing means to detect a logical I/O device access, which is directed to the physical I/O device of the server computer, and a virtual I/O client processing module for transmitting to the server computer, via the network, a command to access the logical I/O device when the logical I/O device access is detected. Awasthi does not disclose this feature.

Another feature of the present invention, as recited in claim 9, includes where the second hypervisor includes a virtual I/O server processing module for receiving the command via the network and for issuing a command to the physical I/O device. Awasthi does not disclose this feature.

Therefore, Awasthi fails to teach or suggest "wherein said first hypervisor comprises: memory protection interrupt processing means called by a memory protection interrupt that is generated when a read or a write occurs to a particular memory address in said first processor, logical I/O device access detecting means that is called by said memory protection interrupt processing means to detect a logical I/O device access, which is directed to said physical I/O device of said server computer, and a virtual I/O client processing module for transmitting to said server computer, via said network, a command to access said logical I/O device when said logical I/O device access is detected" as recited in claim 9.

Furthermore, Awasthi fails to teach or suggest "wherein said second hypervisor comprises a virtual I/O server processing module for receiving said

command via said network and for issuing a command to said physical I/O device" as recited in claim 9.

Each of Crossley, Brice and Awasthi suffer from the same deficiencies, relative to the features of the present invention, as recited in the claims. Therefore, combining the teachings of Crossley, Brice and Awasthi does not render obvious the features of the present invention as now more clearly recited in the claims. Accordingly, allowance of claims 9-14 is respectfully requested.

The remaining references of record have been studied. Applicants submit that they do not supply any of the deficiencies noted above with respect to new claims 9-14.

In view of the foregoing amendments and remarks, Applicants submit that claims 9-14 are in condition for allowance. Accordingly, early allowance of claims 9-14 is respectfully requested.

To the extent necessary, the applicants petition for an extension of time under 37 CFR 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, or credit any overpayment of fees, to the deposit account of MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C., Deposit Account No. 50-1417 (referencing Attorney Docket No. 520.42834X00).

Respectfully submitted,

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.

Donna K. Mason

Donna K. Mason
Registration No. 45,962

DKM/cmd
(703) 684-1120